APPENDIX A

Mode-B Data Transfer Instruction Emulation Sequences

Mode-B Instruction	Mode-A Instruction Sequence	In	Out
MOV #imm,Rn 1110 nnnn siii iiii	1 movi #imm,Rn		
MOV.W @(disp,PC),Rn 1001 nnnn dddd dddd	**1 mova.w disp,R32** 2 ld.w R32,#0,Rn		
MOV.L @(disp,PC),Rn 1101 nnnn dddd dddd	**mova.l disp,R32** 2 ld.l R32,#0,Rn		
MOV Rm,Rn 0110 nnnn mmmm 0011	1 addi Rm,#0,Rn		
MOV.B Rm,@Rn 0010 nnnn mmmm 0000	1 st.b Rn,#0,Rm		
MOV.W Rm,@Rn 0010 nnnn mmmm 0001	1 st.w Rn,#0,Rm		
MOV.L Rm,@Rn 0010 nnnn mmmm 0010	1 st.l Rn,#0,Rm		
MOV.B @Rm,Rn 0110 nnnn mmmm 0000	1 ld.b Rm,#0,Rn		
MOV.W @Rm,Rn 0110 nnnn mmmm 0001	1 ld.w rm,#0,Rn		
MOV.L @Rm,Rn 0110 nnnn mmmm 0010	1 ld.l rm,#0,Rn		
MOV.B Rm,@-Rn 0010 nnnn mmmm 0100	1 st.b Rn,#-1,Rm 2 addi.l Rn,#-1,Rn		
MOV.W Rm,@-Rn 0010 nnnn mmmm 0101	1 st.w Rn,#-1,Rm 2 addi.l Rn,#-2,Rn		
MOV.L Rm,@-Rn 0010 nnnn mmmm 0110	1 strd.w Rn,#-1,Rm 2 addi.l Rn,#-4,Rn		
MOV.B @Rm+,Rn 0110 nnnn mmmm 0100	1 ld.b Rm,#0,Rn 2 ***if (m!=n)*** addi.l Rm,#1,Rm		
MOV.W @Rm+,Rn 0110 nnnn mmmm 0101	1 ld.w Rm,#0,Rn 2 ***if (m!=n)*** addi.l Rm,#2,Rm		
MOV.L @Rm+,Rn 0110 nnnn mmmm 0110	1 ld.l Rm,#0,Rn 2 ***if (m!=n)*** addi.l Rm,#4,Rm		
MOV.B R0,@(disp,Rm) 1000 0000 mmmm dddd	1 st.b Rn,disp,R0		
MOV.W R0,@(disp,Rm) 1000 0001 mmmm dddd	1 st.w Rn,disp,R0		
MOV.L Rm,@(disp,Rn) 0001 nnnn mmmm dddd	1 st.l Rn,disp,Rm		
MOV.B @(disp,Rm),R0 1000 0100 mmmm dddd	1 ld.b Rm,disp,R0		
MOV.W @(disp,Rm),R0 1000 0101 mmmm dddd	1 ld.w Rm,disp,R0		

MOV.L @(disp,Rm),Rn 0101 nnnn mmmm dddd	1 ld.l Rm,disp,Rn		
MOV.B Rm,@(R0,Rn) 0000 nnnn mmmm 0100	1 stx.b Rn,R0,Rm		
MOV.W Rm,@(R0,Rn) 0000 nnnn mmmm 0101	1 stx.w Rn,R0,Rm		
MOV.L Rm,@(R0,Rn) 0000 nnnn mmmm 0110	1 stx.l Rn,R0,Rm		
MOV.B @(R0,Rm),Rn 0000 nnnn mmmm 1100	1 ldx.b Rm,R0,Rn		
MOV.W @(R0,Rm),Rn 0000 nnnn mmmm 1101	1 ldx.w Rm,R0,Rn		
MOV.L @(R0,Rm),Rn 0000 nnnn mmmm 1110	1 ldx.l Rm,R0,Rn		
MOV.B R0,@(disp,GBR) 1100 0000 dddd dddd	1 stx.b R27,disp,R0		
MOV.W R0,@(disp,GBR) 1100 0001 dddd dddd	1 stx.w R27,disp,R0		
MOV.L R0,@(disp,GBR) 1100 0010 dddd dddd	1 stx.l R27,disp,R0		
MOV.B @(disp,GBR),R0 1100 0100 dddd dddd	1 ld.b R27,disp,R0		
MOV.W @(disp,GBR),R0 1100 0101 dddd dddd	1 ld.w R27,disp,R0		
MOV.L @(disp,GBR),R0 1100 0110 dddd dddd	1 ld.l R27,disp,R0		
MOVA @(disp,PC),R0 1100 0111 dddd dddd	1 mova.l disp,R0		
MOVT Rn 0000 nnnn 0010 1001	1 andi R25,#1,Rn		
SWAP.B Rm,Rn 0110 nnnn mmmm 1000	1 byterev Rm,R32 2 shlri Rm,#16,Rn 3 mextr6 R32,Rn,Rn	Rm	
SWAP.W Rm,Rn 0110 nnnn mmmm 1001	1 mperm.w Rm,#1,R32 2 addi.l R32,#0,Rn		
XTRCT Rm,Rn 0010 nnnn mmmm 1101	1 shlli.l Rm,#16,R32 2 shlri.l Rn,#16,Rn 3 or Rn,R32,Rn		

Arithmetic Instruction Emulation Sequences

Mode-B instruction	Mode-A Instruction Sequence	In	Out
ADD Rm,Rn 0011 nnnn mmmm 1100	1 add.l Rm,Rn,Rn		
ADD #imm,Rn 0111 nnnn siii iiii	1 addi.l Rn,#imm,Rn		
ADDC Rm,Rn 0011 nnnn mmmm 1110	1 addz.l Rm,R63,R32 2 addz.l Rn,R63,Rn		

	3 add Rn,R32,Rn 4 add Rn,R25,Rn 5 shlri Rn,#32,R25 6 addi.l Rn,#0,Rn		
ADDV Rm,Rn 0011 nnnn yccc 1111	1 add Rm,Rn,R32 2 add.l Rm,Rn,Rn **3 cmpne Rn,R32,R25**	Rm Rn	
CMP/EQ #imm,R0 1000 1000 siii iiii	1 movi #imm,R32 2 cmpeq R0,R32,R25	R0	
CMP/EQ Rm,Rn 0011 nnnn mmmm 0000	1 cmpeq Rn,Rm,R25	Rm Rn	
CMP/HS Rm,Rn 0011 nnnn mmmm 0010	**1 cmpgeu Rn,Rm,R25**	Rm Rn	
CMP/GE Rm,Rn 0011 nnnn mmmm 0011	**1 cmpge Rn,Rm,R25**	Rm Rn	
CMP/HI Rm,Rn 0011 nnnn mmmm 0110	1 cmpgtu Rn,Rm,R25	Rm Rn	
CMP/GT Rm,Rn 0011 nnnn mmmm 0111	1 cmpgt Rn,Rm,R25	Rm Rn	
CMP/PZ Rn 0100 nnnn 0001 0001	**1 cmpge Rn,R63,R25**	Rn	
CMP/PL Rn 0100 nnnn 0001 0101	1 cmpgt Rn,R63,R25	Rn	
CMP/STR Rm,Rn 0010 nnnn mmmm 1100	1 mcmpeq.b Rm,Rn,R32 2 addz.l R32,R63,R32 3 cmpgtu R32,R63,R25		
DIV0S Rn,Rm 0010 nnnn mmmm 0111	1 xor Rn,Rm,R32 2 ori Rn,#0,R33 3 ori Rm,#0,R33 4 shlri.l R32,#31,R32 5 xori R32,#1,R25 **Q.d = PPF_EX2[63]** 6 nop **M.d = PPF_EX2[63]** **#1.5 dec_br_sr_update = 1**		
DIV0U 0000 0000 0001 1001	1 movi #0,R25 ***Q.d = 0*** ***M.d = 0*** ***#1.5 dec_br_sr_update = 1***		
DIV1 Rm,Rn 0011 nnnn mmmm 0100	1 addz.l Rm,R63,R32 2 ori Rn,#0,R33 ***oldQ.d = Q.q*** 3 shlli.l Rn,#1,Rn 4 addz.l Rn,R25,Rn 5 ***if (oldQ.q == M.q)*** sub Rn,R32,Rn ***lse*** add Rn,R32,Rn ***Q.d = PPF_EX2[63]*** 6 addi.l Rn,#0,Rn ***oldQ.d = Q.q ^ M.q***		

	7 nop 8 nop *Q.d = oldQ.q ^ PPF_EX2[32]* 9 *if (Q.q == 1)* movi #1,R25 *else* movi #0,R25 *#1.5 dec_br_sr_update = 1*		
DT Rn 0100 nnnn 0001 0000	1 addi.l Rn,#-1,Rn 2 cmpeq Rn,R63,R25		
DMULS.L Rm,Rn 0011 nnnn mmmm 1101	1 muls.l Rm,Rn,R24		
DMULU.L Rm,Rn 0011 nnnn mmmm 0101	1 mulu.l Rm,Rn,R24		
EXTS.B Rm,Rn 0110 nnnn mmmm 1110	1 shlli Rm,#56,Rn 2 shari Rn,#56,Rn		
EXTS.W Rm,Rn 0110 nnnn mmmm 1111	1 shlli Rm,#48,Rn 2 shari Rn,#48,Rn		
EXTU.B Rm,Rn 0110 nnnn mmmm 1100	1 andi Rm,#255,Rn		
EXTU.W Rm,Rn 0110 nnnn mmmm 1111	1 shlli Rm,#48,Rn 2 shlri Rn,#48,Rn		
MUL.L Rm,Rn 0000 nnnn mmmm 0111	1 mulu.l Rm,Rn,R32 2 shlri R24,#32,R24 3 mshflo.l R32,R24,R24		
MULS.W Rm,Rn 0010 nnnn mmmm 1111	1 mmullo.wl Rm,Rn,R32 2 shlri R24,#32,R24 3 mshflo.l R32,R24,R24		
MULU.W Rm,Rn 0010 nnnn mmmm 1110	1 shlli Rm,#48,R32 2 shlli Rn,#48,R33 3 shlri R32,#48,R32 4 shlri R33,#48,R33 5 mulu.l R32,R33,R32 6 shlri R24,#32,R24 7 mshflo.l R32,R24,R24		
NEG Rm,Rn 0110 nnnn mmmm 1011	1 sub.l R63,Rm,Rn		
NEGC Rm,Rn 0110 nnnn mmmm 1010	1 addz.l Rm,R63,R32 2 sub R63,R25,Rn 3 sub Rn,R32,Rn 4 shlri Rn,#32,R25 5 addi.l Rn,#0,Rn	Rm	
SUB Rm,Rn 0011 nnnn mmmm 1000	1 sub.l Rn,Rm,Rn		
SUBC Rm,Rn 0110 nnnn mmmm 1010	1 addz.l Rm,R63,R32 2 addz.l Rn,R63,Rn 3 sub Rn,R25,Rn 4 sub Rn,R32,Rn 5 shlri Rn,#32,R25 6 addi.l Rn,#0,Rn		